

OPTIMIZING COMPILATION BY FORWARD STORE MOVEMENT

ABSTRACT OF THE DISCLOSURE

An optimizing compiler includes a component for the determination of potential forward movements of store operations in the compilation of the computer software code. An intermediate representation of computer code is generated including a control flow graph, a data flow graph, a dominator tree, and a reaching defs table. These data structures are accessed to determine where in a conditional branch of code a store operation in the code may be moved to potentially increase efficiency in the execution of the compiled code. Tree structures corresponding to store operations are identified for possible movement, either entirely, or partially. Where a movement of a part of a tree structure is identified, temporary registers may be used to retain values of variables to enable the move to be carried out.